

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Luka Ločniškar

# **Vadba streljanja v navideznem okolju**

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE  
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: prof. dr. Saša Divjak  
SOMENTOR: doc. dr. Matija Marolt

Ljubljana, 2016



Fakulteta za računalništvo in informatiko podpira javno dostopnost znanstvenih, strokovnih in razvojnih rezultatov. Zato priporoča objavo dela pod katero od licenc, ki omogočajo prosto razširjanje diplomskega dela in/ali možnost nadaljne proste uporabe dela. Ena izmed možnosti je izdaja diplomskega dela pod katero od Creative Commons licenc <http://creativecommons.si>

Morebitno pripadajočo programsko kodo praviloma objavite pod, denimo, licenco *GNU General Public License*, različica 3. Podrobnosti licence so dostopne na spletni strani <http://www.gnu.org/licenses/>.

*Besedilo je oblikovano z urejevalnikom besedil L<sup>A</sup>T<sub>E</sub>X.*





Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V okviru diplomske naloge izdelajte sistem za vadbo streljanja v navideznem okolju. Pri tem preučite tehnologije, ki bi se lahko uporabile za sledenje orientacije orožja in izberite ustrezno nizkocenovno rešitev. Izdelajte celovit simulacijski sistem za vadbo streljanja in ga preizkusite.



## IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Luka Ločniškar sem avtor diplomskega dela z naslovom:

*Vadba streljanja v navideznem okolju* (angl. *Shooting practice in a virtual environment*)

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom prof. dr. Saše Divjak in somentorstvom doc. dr. Matija Marolta,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 09. Marca 2016

Podpis avtorja:



# Kazalo

Povzetek

Abstract

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Uvod</b>                              | <b>1</b>  |
| 1.1      | Osnove . . . . .                         | 1         |
| 1.2      | Motivacija . . . . .                     | 2         |
| 1.3      | Analiza stanja v SV . . . . .            | 2         |
| 1.4      | Načrt dela . . . . .                     | 3         |
| 1.5      | Struktura diplomskega dela . . . . .     | 4         |
| <b>2</b> | <b>Razvoj navidezne puške</b>            | <b>5</b>  |
| 2.1      | Analiza obstoječih rešitev . . . . .     | 5         |
| <b>3</b> | <b>Razvoj strelske simulacije</b>        | <b>13</b> |
| 3.1      | Cilj in preverjanje uspešnosti . . . . . | 13        |
| 3.2      | Uporabljena orodja . . . . .             | 13        |
| 3.3      | Razvoj programskega dela . . . . .       | 14        |
| 3.4      | Razvoj strojnega dela . . . . .          | 24        |
| <b>4</b> | <b>Izvedba simulacije in ugotovitve</b>  | <b>25</b> |
| 4.1      | Ugotovitve . . . . .                     | 30        |
| <b>5</b> | <b>Zaključek</b>                         | <b>31</b> |



# Seznam uporabljenih kratic

| kratica     | angleško                                     | slovensko  |
|-------------|--|--|
| <b>SV</b>   | Slovenian army                               | Slovenska vojska                                 |
| <b>MORS</b> | Ministry of Defence,<br>Republic of Slovenia | Ministrstvo za<br>obrambo Republike<br>Slovenije |
| <b>VBS</b>  | Virtual Battlespace                          | Virtual Battlespace                              |
| <b>HMD</b>  | Head mounted display                         | Naglavni ekran                                   |
| <b>API</b>  | Application programming<br>interface         | Vmesnik za programi-<br>ranje aplikacij          |





# Povzetek

**Naslov:** Vadba streljanja v navideznem okolju

Slovenska vojska za usposabljanje vojakov izvaja navidezne in prave treninge. Cilj mojega diplomskega dela je bil implementirati sistem, ki bi omogočal virtualizacijo treninga strelskih vaj. Z določenimi zunanjimi napravami (Razer Hydra) in različnimi programskimi orodji sem razvil sistem, ki omogoča izvedbo strelskih vaj v navideznem okolju. Vojak na treningu s prirejeno plastično puško meri in strelja v ekran, ki prikazuje tarče in po izvedenem treningu analizira kvaliteto strelca.

Rešitev sem evalviral z izvedbo strelske vaje z implementiranim sistemom in rezultate prikazal v poglavju 4.

**Ključne besede:** streljanje, Razer Hydra, simulacija, Unity.



# Abstract

**Title:** Shooting practice in a virtual environment The Slovenian army trains its soldiers through various virtual and field training programs. The goal of my thesis was to implement a system for the virtualization of a shooting range practice. Using certain input devices (Razer Hydra) and different software tools I implemented a system which allows for shooting practice to be executed in a virtual environment. The trainee uses a modified plastic rifle to aim and shoot at a screen which displays the training simulation. I evaluated the system by executing a shooting practice session. The results are analyzed in chapter 4.

**Keywords:** shootig, Razer Hydra, simulation, Unity.



# Poglavje 1

## Uvod

### 1.1 Osnove

Z napredkom tehnologije se pojavljajo možnosti za izboljšave in modernizacijo postopkov, ki niso direktno povezani z računalniško znanostjo. Ta pojav ni nič novega, saj je, vsaj v zahodnem svetu, razvoj novih tehnologij tesno povezan z razvojem novih orodij in orožij za uporabo v vojskah.[1] Pogosto so nove tehnologije razvite primarno za uporabo v vojaških postopkih in se kasneje komercializirajo in uporabijo v produktih namenjenih potrošnikom. Primer takega produkta je avtomatski sesalec iRobot Roomba. Podjetje, ki je razvilo zelo popularen sesalec, je pred migracijo v svet masovne potrošnje isto tehnologijo uporabljalo za robote uporabljene v vojaških spopadih. Roboti so bili namenjeni raziskovanju nevarnih območij in detonacijo min. Obstajajo pa tudi obratni primeri, kjer je bila določena tehnologija najprej razvita za navadne potrošnike in kasneje uporabljena v vojskah. Primer takega produkta je VBS, Virtual Battlespace. Po uspehu prvoosebni strelskih računalniških iger se je pojavila želja po simulaciji vojaških spopadov z namenom usposabljanja in treniranja vojakov. Želji je ustreglo podjetje Bohemia Interactive, ki je razvilo omenjeni simulator z naslovom VBS (ali Virtual Battlespace, najnovejša verzija VBS 3).

## 1.2 Motivacija

Z napredkom tehnologije se spreminja način vojskovanja in treniranja vojakov. Ena od novih metod treniranja vojakov je navidezna strelska simulacija. [1] [3]

Obstoječe rešitve za navidezne treninge streljanja temeljijo na izjemno dragih laserskih sistemih. Zato sem si kot cilj zastavil izdelavo simulatorja streljanja z dostopnimi tehnologijami.

S pomočjo teh tehnologij sem poskušal dokazati, da je urjenje vojakov v veščini streljanja mogoče izpeljati na lažji in cenejši način. Strelesko vajo sem poskušal pripeljati v zaprte prostore, brez potrebe po rabi streliva in s hitrejšim, bolj natančnim ocenjevanjem učinkovitosti in natančnosti.

## 1.3 Analiza stanja v SV

Analizirali smo orodja ki jih uporablja SV in ugotovili, da SV aktivno uporablja VBS 2 za treniranje njihovih vojakov.

VBS je zelo široko, realistično in podrobno navidezno simulacijsko okolje, ki omogoča trening vojakov iz udobja pisarniškega stola. Vojake primarno urijo v njihovih sposobnostih sprejemanja odločitev v zelo nevarnih in stresnih situacijah. VBS 3 med drugimi uporablja tudi NATO, ki ima zakupljenih nekaj licenc simulacije in jih oddaja obrambnim silam svojih članic. [2] Zaradi uspešnosti te simulacijske programske opreme je licenco kupilo tudi MORS in namreč VBS verzije 2.

A sam trening v simuliranem okolju zahteva, da vojak sedi za (stari) računalnikom ter s standardnimi vhodnimi napravami (tipkovnica in miška) upravlja svoj navidezni avatar. Vsak njegov premik in vsaka njegova odločitev se beležita in pošljeta nadrejenemu, odgovornemu za trening. Pravilne odločitve se nato nagradijo, napačne pa sankcionirajo. Predstavniki SV so izrazili nezadovoljstvo s samo tehnologijo pri interakciji med vojakom, trenirancem in računalnikom. Trening v VBS vojaka ne postavi v dovolj realno situacijo, predvsem zaradi načina interakcije z računalnikom. Vojak

na treningu ni izpostavljen stresu, nevarnosti in strahu, ki so prisotni v vsakem vojaškem spopadu. Predstavniki SV so zato izrazili željo po sistemu, ki bi trening na navideznem bojišču bolj približal realni situaciji vojaških spopadov.

## 1.4 Načrt dela

Postavili smo načrt za izpeljavo želja SV. Načrt sistema za bolj realistično simulacijo smo razdelili na naslednje dele:

- Bolj realističen pogled v navidezno simulacijo
  - Namesto, da vojak dobiva informacijo o svoji navidezni okolici in povratno informacijo svojih dejanj skozi monitor, bi prikaz simulacije prenesli na eno od naprav HMD za navidezno resničnost
    - \* Oculus Rift
    - \* Morpheus
    - \* Gear VR
- Premikanje vojaka v realnem svetu
  - Namesto premikanja navideznega avatarja preko tipkovnice bi omogočili, da se vojak premika oziroma izvaja določene gibe v realnem svetu, ki bi se prevedli v gibe/premikanje v navideznem svetu. Te gibe bi zaznali z neko zunanjo napravo in jih analizirali.
    - \* Kinect 2.0
- Merjenje in streljanje s puško
  - Namesto merjenja in streljanja z računalniško miško bi vojaku leto omogočili z bolj napredno napravo, ki bi bila boljši približek realnosti.

V sklopu svoje naloge sem se osredotočil na zadnjo točko.

Za izpeljavo bi uporabil kak obstoječ sistem ali pa implementiral svojega.

Po implementaciji bi izpeljal simulacijo, ki bi dokazala ali ovrgla uspešnost razvitega sistema.

## 1.5 Struktura diplomskega dela

Diplomsko delo je logično razdeljeno na naslednje dele:

- Analiza obstoječih rešitev za navidezne strelske simulacije
- Razvoj programskega dela simulacije
- Razvoj strojnega dela simulacije
- Načrtovanje in izvedba testiranja



## Poglavje 2

# Razvoj navidezne puške

### 2.1 Analiza obstoječih rešitev

Da bi lahko razvil čim bolj natančen in odziven sistem za merjenje s puško, sem naredil analizo tehnologij, ki so na voljo. Te tehnologije so v večini namenjene komercialni uporabi. Cilj je bil najti čim bolj natančno in čim cenejšo tehnologijo za implementacijo načrtovanih funkcionalnosti.

#### 2.1.1 Sledenje gibanja s kamero in označevalnikom

Ta tehnologija uporablja dva ključna elementa. Eden je oddajnik (ponavadi na upravljavcu), drugi pa sprejemnik (zunanja kamera ali kamera v konzoli). označevalnik vsebuje objekt določene oblike (ponavadi krogle), ki ima LED svetila in se pri uporabi sveti z določeno barvo. Barva svetila je odvisna od barve okolice, da je prejemniku (kamero) lažje ločiti označevalnik od samega ozadja. Ko se označevalnik in kamera sinhronizirata, začne kamera zajemati sliko prostora in z informacijo o barvi označevalnika določi pozicijo označevalnika na ravnini. Ravnino v prostoru izračuna glede na velikost označevalnika v zajeti sliki (večji kot je označevalnik, bližje je igrallec z upravljavcem). Kot dodatno informacijo pa upravljavec vsebuje tudi pospeškometer za potrebe dodatnih funkcionalnosti in večjo natančnost.

Lastnosti takega sistema:



Slika 2.1: Primer takega sistema (PS Move)

- + Dokaj poceni
- + Dostopna tehnologija
- + Dobra odzivnost
- Slaba natančnost
- Upravljalvec mora biti konstanto obrnjen proti kameri

### 2.1.2 Infrardeča svetlobna puška

Ta način simulacije prav tako uporablja oddajnik svetlobe in nek prejemnik, ki zazna določeno frekvenco (oddajnikove) svetlobe. Prejemnik (kamera) informacijo o strelu zajema v realnem svetu in jo glede na orientacijo kamere in pozicijo projekcije (ali televizije) preslika v navidezni svet. Preproste in manj



Slika 2.2: Primer puške s takim sistemom (Trinity Magnum)

natančne sisteme se da implementirati kar s preprostim laserskim pisalom in navadno spletno kamero. Na voljo so tudi odprtokodne implementacije takih sistemov. Profesionalni sistemi za simulacije streljanja pa uporabljajo specializirane oddajnike in prejemnike svetlobe za visoko natančnost. [4]

Lastnosti takega sistema:

- + Visoka natančnost
- + Pogosto uporabljena za namen strelskih simulacij
- + Dobra odzivnost
- Visoka cena sistema
- Upravljevec mora biti obrnjen v isto smer kot kamera

Obstaja tudi obratni primer obravnavanega sistema, kjer je oddajnik kamera, prejemnik pa individualni upravljevec. Primer takega sistema je komercialno uspešen sistem Nintendo Wii. [5] Ta sistem deluje tako, da statični



Slika 2.3: LaserShot strelski simulator

del (Wii bar), ki je postavljen na televizijo, v prostor spušča infrardeče žarke, ki so človeškemu očesu nevidni. Žarke zajame vsak upravljavec, sinhroniziran z Wii konzolo, in preko bluetooth protokola sami konzoli pošlje informacijo o orientaciji.

Lastnosti takega sistema:

- + Dobra natančnost
- + Nizka cena
- Natančnost ni dovolj visoka za strelske simulacije
- Upravljavec mora biti obrnjen proti Wii bar-u

### 2.1.3 Simulirana puška z uporabo informacije o poziciji upravljavca in o vidnem polju uporabnika

S popularizacijo navidezne resničnosti se je pojavila potreba po vhodnih napravah, za katere ni potrebno, da so stalno usmerjene proti prejemniku (kameri). Naprave za prikaz igre oziroma simulacije pri navidezni resničnosti



Slika 2.4: LaserShot strelski simulator

namreč niso na statični poziciji v prostoru, temveč so (najbolj pogosto) pritrjene na glavo uporabnika in se skupaj z uporabnikom primerno premikajo in rotirajo.

Take vhodne naprave uporabljajo informacijo o absolutni poziciji in rotaciji upravljavca ter o vidnem polju uporabnika in tako določijo, kam uporabnik meri svoj upravljavec. Večina takih naprav je še v fazi razvoja. Najbolj odmeven je produkt Stem System podjetja Sixense. Ta sistem deluje s pomočjo šibkega magnetnega polja, ki ga ustvarja centralna postaja (baza) sistema. Obstaja tudi bolj zgodnja iteracija te naprave, ki jo je podjetje Sixense razvilo v povezavi s podjetjem Razer. Imenuje se Razer Hydra. Ta naprava primarno ni bila namenjena za uporabo v navidezni resničnosti, saj v času njenega razvoja navidezna resničnost še ni bila trend v računalniški grafiki in igričarski industriji. Razer Hydra je za razliko od Stem Systema povezana do glavne postaje (baze) s kablom in ne vsebuje dodatnih označevalnikov za določanje pozicije in orientacije samega uporabnika. [6]

Lastnosti takega sistema:

- + Visoka natančnost (na 1 mm pri poziciji in 1° pri orientaciji)

- + Nizka cena
- + Dobra odzivnost
- + Neodvisnost smeri merjenja
- Občutljivost sistema na motnje (kovina moti magnetno polje)

Zaradi naštetih lastnosti sem se odločil, da bo ta sistem najbolj primeren za implementacijo zastavljenih funkcionalnosti. Razlog za tako izbiro je tudi v tem, da sistem za navidezno resničnost zagotavlja neodvisnost smeri merjenja naprave, ki pa je nujni pogoj sistema za vojaško simulacijo. Tako vse druge možnosti odpadejo saj zahtevajo uporabo sprejemne naprave, ki omeji smer merjenja ekran ali steno, kjer se predvaja simulacija oziroma igra.



Slika 2.5: LaserShot strelski simulator



Slika 2.6: StemSystem



Slika 2.7: Razer Hydra



## Poglavje 3

# Razvoj strelske simulacije

### 3.1 Cilj in preverjanje uspešnosti

Cilj razvoja je bil dokazati ali ovreči učinkovitost in natančnost implementiranega sistema. S tem namenom sem izvedel simulacijo navidezne strelske vaje. Vaja je bila izvedena po standardih strelskih vaj MORS.

### 3.2 Uporabljena orodja

#### 3.2.1 Unity pogon za igre

Kot razvijalsko okolje za implementacijo strelskega sistema sem uporabil pogon za igre Unity. Razlogov za to je mnogo, a glavna sta dostopnost in že implementiran API za delo z Razer Hydro. Unity omogoča hitro implementacijo in zelo kvaliteten okvir za prototipiranje iger in simulacij. Podjetje Sixense je kot pobudo za razvoj iger z njihovimi napravami zastonj objavilo API, ki procesira podatke njihovih vhodnih naprav.

#### 3.2.2 Razer Hydra

Razer Hydra je vhodna naprava primarno razvita z namenom uporabe v prvoosebni strelski igrah. Naprava je sestavljena iz treh komponent: dva

upravljavca in ena baza, ki ustvarja šibko magnetno polje. Kot motnji v tem polju vsak od upravljavcev lahko izračuna svojo pozicijo in rotacijo v prostoru (glede na bazo). Po ugotovitvah analize orodji za implementacijo strelske simulacije sem zaradi razlogov naštetih v prejšnjem poglavju, izbral napravo Razer Hydra. Ko bo dokončno razvit Stem System in bo na voljo za prodajo Razer Hydra zastarela naprava, ki po vsej verjetnosti ne bo več masovno uporabljena. Ker oba sistema delujeta na istem principu (magnetno polje), bo strelski simulator deloval tudi z novo napravo.

## 3.3 Razvoj programskega dela

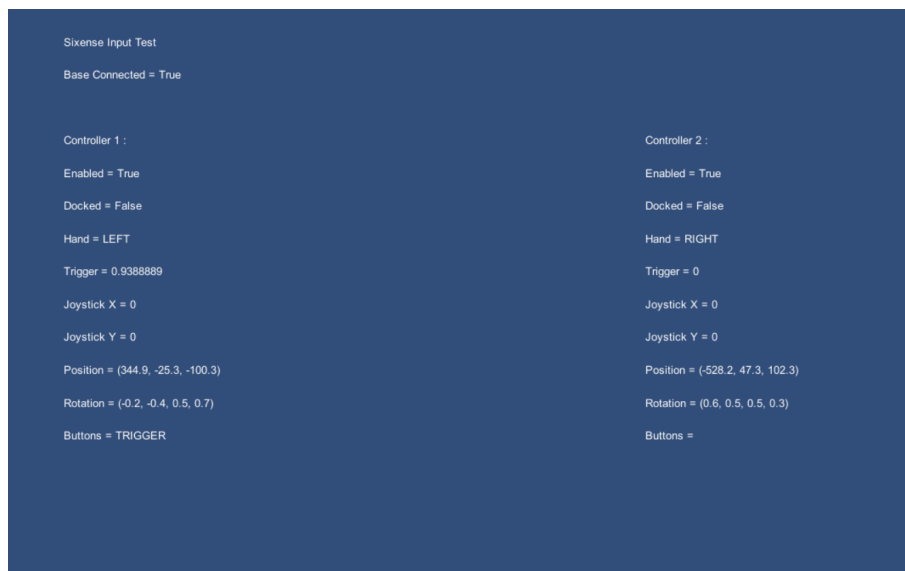
### 3.3.1 Procesiranje vhoda

Igre, ki za vhodno napravo uporabljajo Razer Hydro, uporabljajo le enega od dveh upravljavcev, ki sta na voljo pri napravi. Drugi upravljavec pa se uporablja v namen premikanja lika z analogno gobico in ostalimi gumbi.

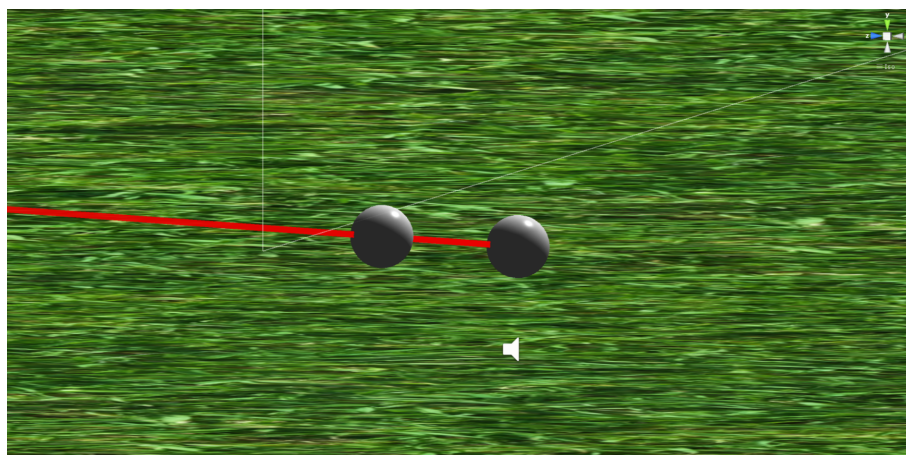
Za doseganje večje natančnosti sem implementiral sistem, ki za merjenje z napravo uporablja oba upravljavca. API, ki sem ga uporabil za zajem vhodne naprave, za vsako sliko simulacije (v povprečju 60 na sekundo) prejme od vsakega upravljavca določene podatke:

- Pozicijo upravljavca v prostoru (s koordinatami  $x,y,z$ , kjer  $0,0,0$  predstavlja bazo Razer Hydre).
- Rotacijo upravljavca
- Spremembe vseh gumbov upravljavca

Smer merjenja je tako določena s poltrakom z izvorom v točki pozicije enega od upravljavcev, ki poteka skozi točko določeno s pozicijo drugega upravljavca. Za vsako sliko izrisa se izračuna vektor `fireDir`, ki predstavlja normaliziran vektor v smeri merjenja.



Slika 3.1: Procesiranje vhoda Razer Hydra



Slika 3.2: označevalnika upravljavcev in smer merjenja

Listing 3.1: Koda za pridobitev pozicij obeh upravljavcev (C#)

```
if (hand.m_hand == SixenseHands.RIGHT) {  
    position1 = hand.transform.position;  
} else if (hand.m_hand == SixenseHands.LEFT) {  
    position2 = hand.transform.position;  
}  
fireDir = Vector3.Normalize (position2 - position1);
```

S tem načinom merjenja je sistem dosegal že dokaj dobre rezultate z vidika natančnosti. Kljub temu so bile zelo opazne motnje in tresljaji, ko sta bila upravljavca v stanju mirovanja (statična). Zato sem zajel in izrisal podatke o poziciji upravljavcev. Na ta način so postala opazna odstopanja od samih premikov upravljavcev. Pokazal se je eden večjih problemov naprave Razor Hydra in sicer motnje iz okolice. Večja kovinska telesa v bližini baze Razer Hydre so povzročala šum v signalu. Signal je bilo treba narediti bolj odporen na zunanje motnje in ga gladiti.

### 3.3.2 Filtriranje signalov vhoda

Filtriranje je v procesiranju signalov definirano kot odstranjevanje ali omejevanje določenih nezaželenih aspektov signala. V našem primeru so to motnje pri premikanju upravljavcev. Kot kaže slika 3.3, so odstopanja od gladkega, nemotenega gibanja očitna.

Za filtriranje podatka o poziciji upravljavca sem implementiral Kalmanov filter.

#### Kalmanov filter

**Splošno** "Kalmanov filter je algoritem, ki uporablja serijo meritev, zajetih skozi čas, ki vsebujejo statistični šum in druge netočnosti, in proizvede ocene neznanih spremenljivk, ki so pogosto bolj natančne kot tiste, ki bazirajo na posameznih meritvah." [8]

Preprosto povedano, Kalmanov filter ob vsaki novi meritvi (v našem primeru pozicije upravljavca) z upoštevanjem dosedanjih meritev aproksimira novo stanje, tako da so meritve na splošno bolj točne.

**Matematična definicija** Kalmanov filter temelji na linearnem dinamičnem sistemu v domeni diskretiziranega časa. Modeliran je po Markovskih verigah. Stanje sistema predstavlja vektor realnih števil. V vsakem časovnem inkreментu se stanje spremeni z nekim linearnim operatorjem, ki vsebuje šum. [7]

Stanje Kalmanovega filtra je določeno z:

- $x_{k|k}$  predvideno stanje ob času  $k$  pri danih vseh vhodih od prvega do  $k$
- $P_{k|k}$  kovariančna matrika napake

Da lahko iz vhodov, ki vsebujejo šum, predvidimo stanje procesa, moramo proces modelirati z upoštevanjem okvira Kalmanovega filtra. Določiti je potrebno naslednje matrike. [7]

- $F_k$  model tranzicije med stanji
- $H_k$  opazovalni model
- $Q_k$  kovarianca šuma procesa
- $R_k$  kovarianca opazovalnega šuma
- (včasih)  $B_k$  model kontrolnega vhoda, za vsak časovni inkrement ( $k$ )

S Kalmanovim filtrom predvidmo pravo stanje (ob času  $k$ ) iz prejšnjega stanja (ob času  $k - 1$ ) z enačbo:

$$x_k = F_k x_{k-1} + B_k u_k + q_k \quad (3.1)$$

kjer je:  $q_k$  šum procesa, dobljen iz multivariate normalne porazdelitve s kovarianco  $Q_k$ .

Ob času  $k$  je tako vhod  $z_k$  pravega stanja  $x$  izračunan tako:

$$z_k = H_k x_k + r_k \quad (3.2)$$

kjer je:  $r_k$  opazovani šum, ki je povprečna vrednost Gaussovega belega šuma s kovarianco  $R_k$ .

Kalmanov filter poteka v dveh fazah: napoved in posodobitev. V fazi napovedi se z uporabo predvidenega stanja prejšnjega koraka ( $k - 1$ ) izračuna predvideno stanje trenutnega koraka ( $k$ ). V fazi posodobitve pa se uporabi predvideno stanje faze napovedi in informacija o vhodu. [7]

**Napoved**

$$\begin{aligned}\hat{x}_{k|k-1} &= F_k \hat{x}_{k-1|k-1} + B_k u_k \\ P_{k|k-1} &= F_k P_{k-1|k-1} F_k^T + Q_k\end{aligned}\tag{3.3}$$

**Posodobitev**

$$\begin{aligned}\tilde{y}_k &= z_k - H_k \hat{x}_{k|k-1} \\ S_k &= H_k P_{k|k-1} H_k^T + R_k \\ K_k &= P_{k|k-1} H_k^T S_k^{-1} \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k \tilde{y}_k \\ P_{k|k} &= (I - K_k H_k) P_{k|k-1}\end{aligned}\tag{3.4}$$

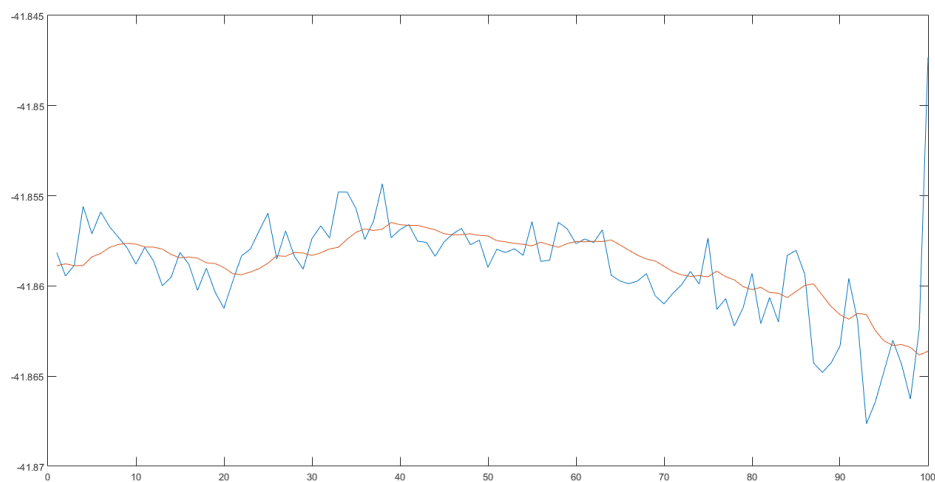
**Implementacija za naš primer****Predpostavka**

$$\begin{aligned}x &= \text{filtrirana meritev} \\ z &= \text{meritev} \\ q &= \text{šum procesa} \\ r &= \text{šum senzorja} \\ p &= \text{predvidena napaka} \\ k &= \text{Kalman Gain}\end{aligned}\tag{3.5}$$

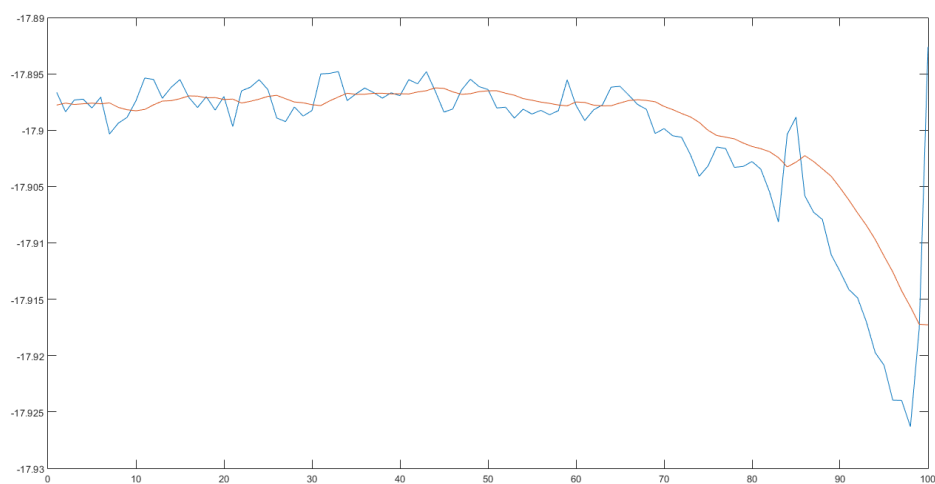
**Izračun nove filtrirane meritve**

$$\begin{aligned}p &= p + q \\ k &= p / (p + r); \\ x &= x + k * (z - x) \\ p &= (1 - k) * p\end{aligned}\tag{3.6}$$

Po filtriranju vhoda dobimo mnogo bolj gladek signal oziroma informacijo o poziciji upravljavca.

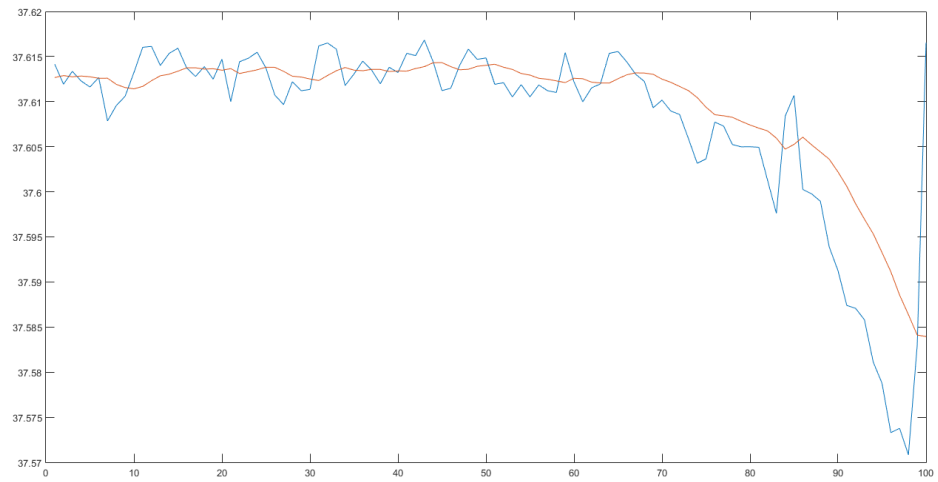


Slika 3.3: Nefiltriran(modra) in filtriran(oranžna) vhod upravljavca, koordinata x

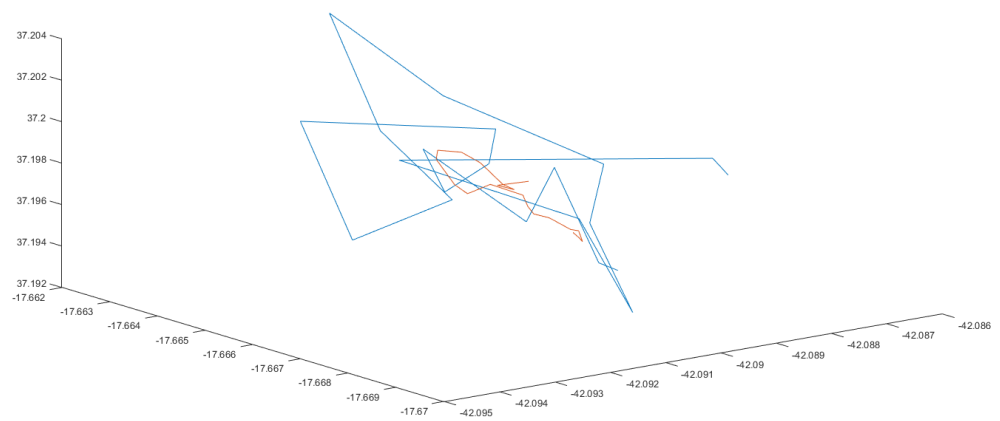


Slika 3.4: Nefiltriran(modra) in filtriran(oranžna) vhod upravljavca, koordinata y





Slika 3.5: Nefiltriran(modra) in filtriran(oranžna) vhod upravljavca, koordinata z



Slika 3.6: Nefiltriran(modra) in filtriran(oranžna) vhod upravljavca, 3D

### 3.3.3 Preslikava v navidezni svet

V tej točki je merjenje z napravo zelo natančno in s pomočjo filtriranja s Kalmanovim filtrom dokaj gladko. A cilj sistema je uporabniku omogočiti merjenje direktno na ekran, kjer se predvaja simulacija. Za doseg tega cilja je bilo treba ugotoviti, kje v prostoru se nahaja ekran, na katerega uporabnik meri, in to preslikati v navidezni svet. To sem dosegel z implementacijo posebne kalibracije.



Slika 3.7: Scena strelske vaje

#### Kalibracija

Pred streljanjem mora uporabnik vnesti, na kakšni razdalji od strelca je ekran, kjer se predvaja simulacija. Nato uporabnik nameri v robove ekrana in stisne sprožilec na upravljavcu Razer Hydre. V simulaciji se ob vsakem pritisku izračuna presek med poltrakom, ki ga določa uporabnik s smerjo merjenja, in ravnino, na kateri leži ekran (prikaz simulacije). Na ravnini se nato glede na izračunane točke ustvari pravokotnik, ki določa, kam lahko uporabnik meri.

Bolj natančno kalibracija poteka v naslednjih korakih:

- Inicializacija in določitev ravnine ekrana. Ravnina v prostoru je določena

s točko na ravnini in normalo ravnine.

- Normala je vektor, določen z orientacijo kamere. V simulaciji je to 3D vektor  $\vec{n}(0, 0, 1)$
- Točka na ravnini je predstavljena s 3D vektorjem  $\vec{v}_t(0, 0, d_e)$ , kjer je  $d_e$  razdalja od strelca do ekrana.
- Iz teh podatkov lahko izračunamo enačbo ravnine z  $n_x(x - x_v t) + n_y(y - y_v t) + n_z(z - z_v t) = 0$  in dobimo  $z - d_e = 0$
- Določitev kontrolnih točk ekrana na ravnini. Za bolj natančno kalibracijo sem uvedel 8 kontrolnih točk, ki so:
  - Zgoraj (levo, center, desno)
  - Sredina (levo, desno)
  - Spodaj (levo, center, desno)

Z vsakim pritiskom na sprožilec se izračuna presek med smerjo merjenja  $fireDir$  in ravnino, določeno v prvem koraku, na sledeči način:

$$\begin{aligned}
 top &= \vec{n} \cdot (\vec{v}_s - \vec{v}_t) \\
 bottom &= \vec{n} \cdot fireDir \\
 t &= -(top/bottom) \\
 točka &= \vec{v}_s + t fireDir
 \end{aligned} \tag{3.7}$$

Kjer je  $\vec{n}$  normala ravnine,  $\vec{v}_s$  točka izvora merjenja (pozicija puške),  $\vec{v}_t$  točka na ravnini,  $fireDir$  vektor, ki določa smer merjenja.

- Določitev centra pravokotnika (ekrana) na ravnini in izračun njegove velikosti Z zadnjim (devetim) pritiskom na sprožilec strelec določi center pravokotnika ekrana. Izračunan je z enačbo (3.7). Širina in višina pravokotnika se določita s povprečjem osmih kontrolnih točk iz prejšnjega koraka.

Med izvajanjem simulacije se vsako sliko (frame) izračuna presek med smerjo merjenja in ravnino ekrana z enačbo (3.7). Če je točka preseka znotraj izračunanega pravokotnika uporabnik meri na ekran. Ko uporabnik ustreli (pritisne sprožilec), se iz točke preseka v sceno vrže žarek (Raycast), ki predstavlja sam strel puške. Po kalibraciji tako uporabnik v simulacijo gleda skozi okvir (pravokotnik) določen s kalibracijo.

### 3.4 Razvoj strojnega dela

Da bi bila simulacija bližje realnosti sem sestavil plastično puško, ki vsebuje upravljalce Razer Hydra.



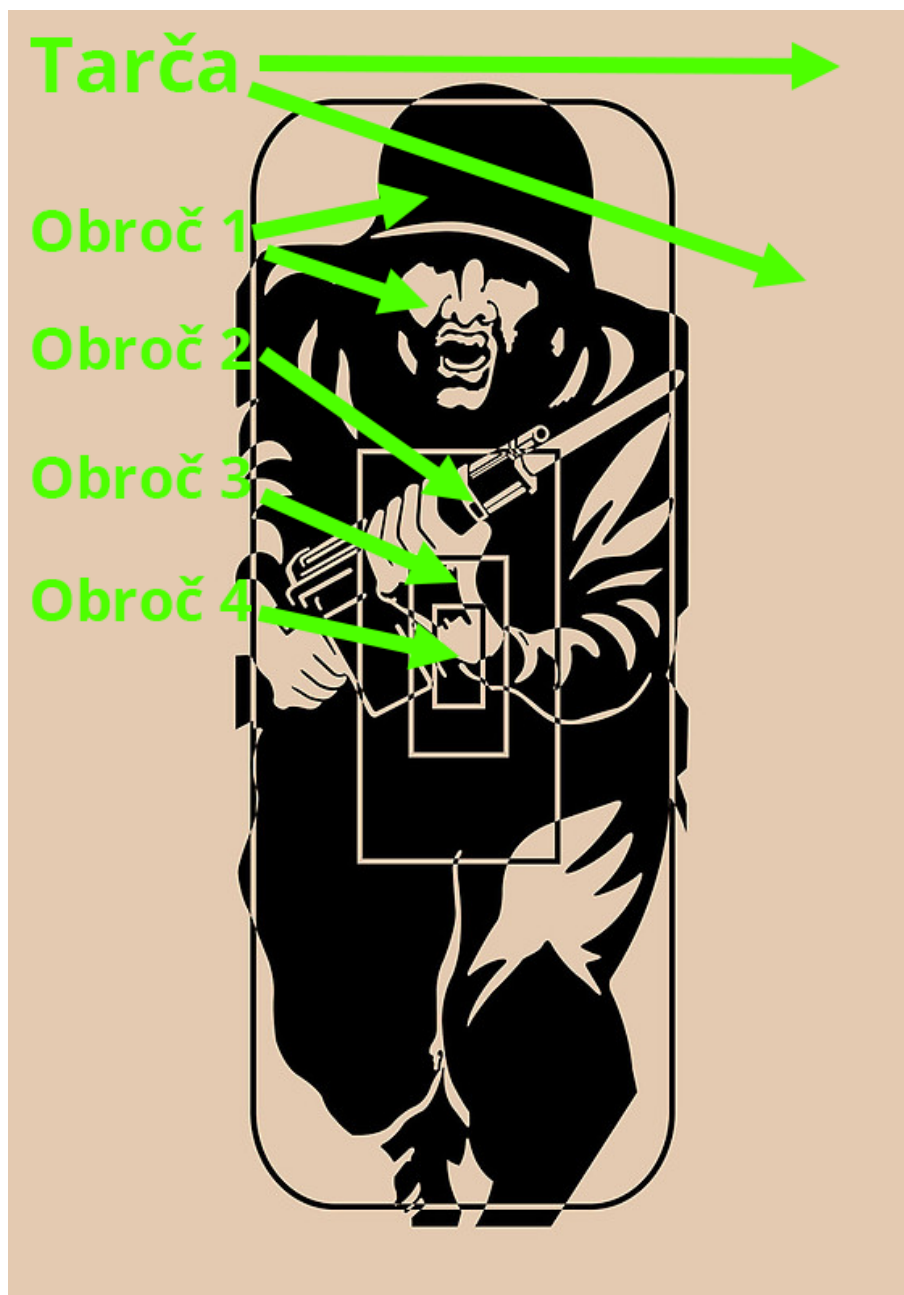
Slika 3.8: Razer Hydra Puška

Okvir puške je iz razstavljene airsoft puške. Vgrajene ima čipe z razstavljenih upravljalcev Razer Hydra. Pri sestavljanju puške sem pazil, da sta čipa čim bolj narazen zaradi doseganja čim večje natančnosti pri merjenju.

## Poglavje 4

# Izvedba simulacije in ugotovitve

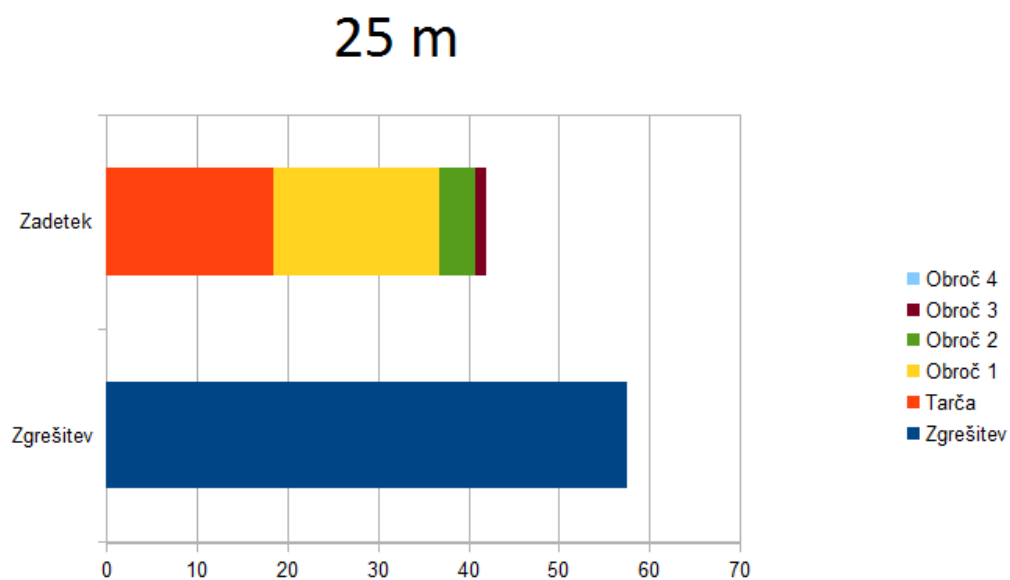
Da bi dokazal ali ovrgel uporabnost sistema, sem na FRI organiziral testiranje simulacije. Testiranja se je udeležilo 50 udeležencev, od katerih je vsak izvedel kalibracijo in nato poskušal čim bolj uspešno izvesti strelsko vajo. Streloske vaje so bile določene s standardi SV in MORS. Udeleženci so iz sedečega položaja streljali na tarče oddaljene 25m, 100m in 200m. Tarča je, kot je prikazano na sliki 4.1, razdeljena na pet delov. Rezultat vsakega strela je lahko zadetek ali zgrešitev. Zadetek pa se še deli, glede na lokacijo zadetka na: tarča, obroč 1, obroč 2, obroč 3 in obroč 4, kjer je najlažje zadeti tarčo, najtežje pa obroč 4. Grafi 4.3, 4.4, 4.5 prikazujejo (v %) deleže zadetkov (hit) in zgrešitev (miss) vseh udeležencev testiranja.



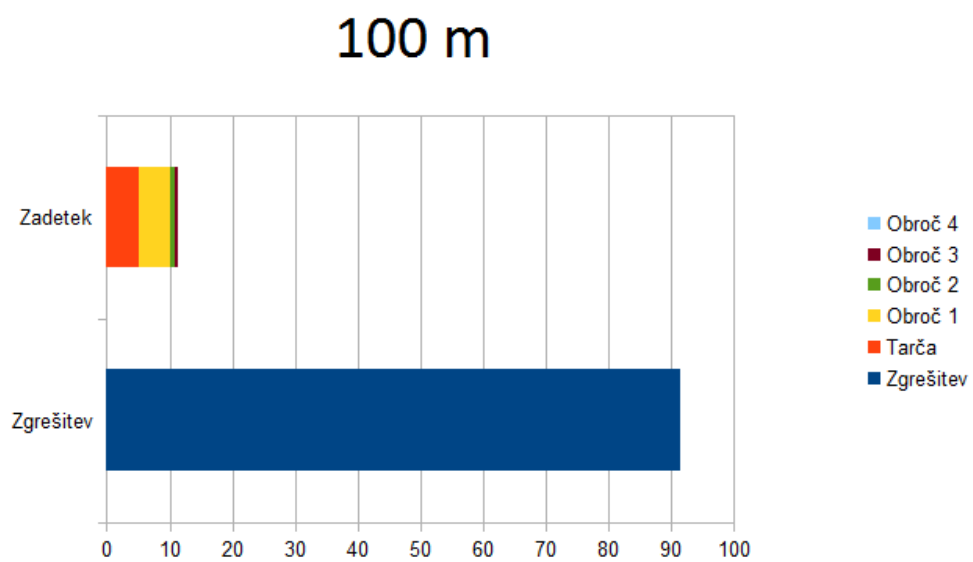
Slika 4.1: Tarča stelske simulacije



Slika 4.2: Slika simulacije v izvajanju

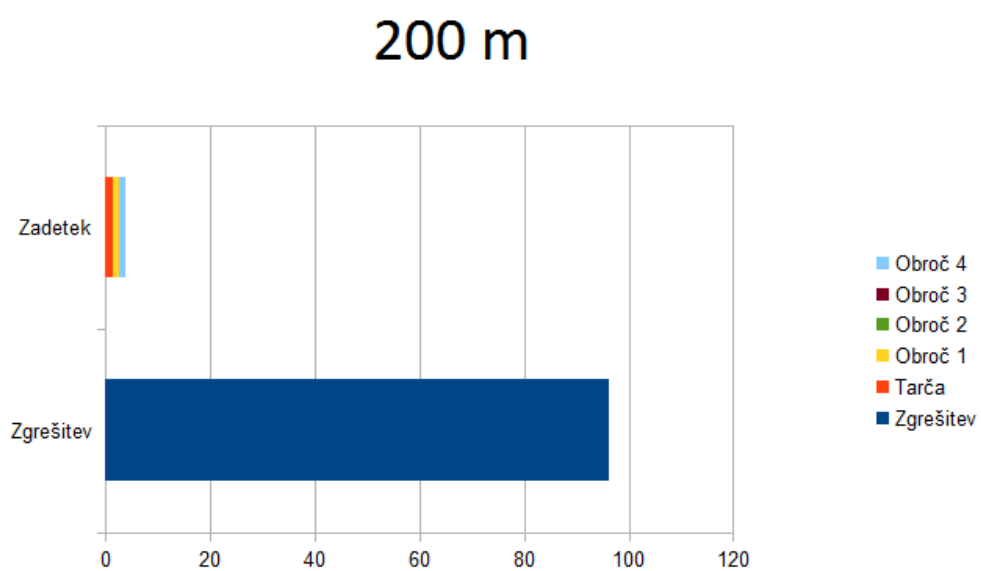


Slika 4.3: Delež zadetkov proti zgrešitvam pri tarčah na 25m



Slika 4.4: Delež zadetkov proti zgrešitvam pri tarčah na 100m





Slika 4.5: Delež zadetkov proti zgrešitvam pri tarčah na 200m

## 4.1 Ugotovitve

Večina udeležencev se je pri streljanju zanašala zgolj na povratno informacijo prejšnjega strela, da so tarčo zadeli. Ko je sodelujoči stisnil sprožilec, se je na ekranu prikazal žarek, ki je pokazal, kam je strelec ustrelil. Namesto da bi udeleženci merili čez merilec puške, so ugotovili, da je tarčo mnogo lažje zadeti, če puško premikajo glede na lokacijo prejšnjega zadetka. Poskušal sem poustvariti pristno strelsko vajo, kjer bi se strelec zanašal na svoje sposobnosti merjenja in veščine in ta način streljanja tega ne odraža.

Po analizi rezultatov testiranja in pogovoru z udeleženci sem prišel do zaključka, da sistem ni dovolj natančen za strelsko simulacijo. Razlog za to se nahaja v sami tehnologiji Razer Hyde. Čeprav je upravljavec v zelo kontroliranih pogojih zanesljiv, vsaka motnja ali sprememba sistema povzroči odstopanja od smeri merjenja in lokacije zadetka. Motnje nastopijo v oblikah:

- Veliki kovinski predmeti v prostoru testiranja
- Kakršnikoli premiki baze po izvedeni kalibraciji

Da se pri strelski vaji res pokažejo veščine strelca, mora biti razlika med smerjo, kamor meri, in kam ustrelji minimalna (milimetri). Čeprav se to z implementiranim sistemom do nekega nivoja da doseči, je sistem preveč občutljiv na spremembe, da bi bil uporaben kot pravi strelski simulator.

## Poglavje 5

### Zaključek

Razvita vadba je premalo natančna za preverjanje natančnosti strelca zaradi upravljavcev Razer Hydra. Če pa bi za izvedbo uporabljal bolj natančne senzorje bi bila vadba, z vsemi razvitimi komponentami, primerna za trening streljanja.

Za trenutno implementacijo pa bi se še vedno dalo najti bolj primerno uporabo, kjer visoka natančnost ni ključnega pomena. Primer take uporabe je recimo simulator za gašenje požarjev ali pa kakšno varjenje.

Implementiran sistem pa je dober za uporabo v navidezni resničnosti, kjer sam prikaz (ekran) simulacije ni statičen, temveč se premika skupaj s strelcem. Razlog za večjo uporabnost v navidezni resničnosti je to, da smer merjenja in samo streljanje ni treba preslikati iz realnega v navidezni svet, saj se celotna simulacija izvaja v navideznem svetu. V tem primeru tudi kalibracija ni potrebna.

Vadba je dobra podlaga za nadaljnji razvoj in vključitev v celoten sistem navideznega treninga. Tudi v polni simulaciji implementirana rešitev ne bo odražala kvalitete strelca, bo pa izkušnjo vseeno približala realnosti.



# Literatura

- [1] J. D. Fletcher, “Education and Training Technology in the Military”, *Science*, št. 323, zv. 5910, str. 72–75, 2009.
- [2] NATO, [Online], VBS3 NATO. Dosegljivo:  
<https://nato.bisimulations.com/>. [Dostopano 10.1.2016]
- [3] T. Farrell, T. Terriff, “The sources of military change”, Lynne Rienner Publishers , 1970.
- [4] Wikipedia [Online], Dosegljivo:  
[http://en.wikipedia.org/wiki/Light\\_gun](http://en.wikipedia.org/wiki/Light_gun), [Dostopano 15.1.2016].
- [5] B. Marshall, How the Wii Works [Online]. Dosegljivo:  
<http://electronics.howstuffworks.com/wii2.htm>. [Dostopano 15.1.2016].
- [6] P. Miller, Razer Hydra motion controller hands-on [Online]. Dosegljivo:  
<http://www.theverge.com/2011/06/09/razer-hydra-motion-controller-hands-on>. [Dostopano 15.1.2016].
- [7] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems”, *Journal of Basic Engineering*, str. 82–35, 1960.
- [8] Wikipedia [Online], Dosegljivo:  
[https://en.wikipedia.org/wiki/Kalman\\_filter](https://en.wikipedia.org/wiki/Kalman_filter), [Dostopano 10.12.2015].